UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/726,144	11/29/2000	Frank Worrell	00-419 1496.00055	6557

24319 7590 03/29/2004

LSI LOGIC CORPORATION
1621 BARBER LANE
MS: D-106 LEGAL
MILPITAS, CA 95035

EXAMINER	
GERSTL, SHANE F	
ART UNIT	**PAPER NUMBER**
2183	

DATE MAILED: 03/29/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/726,144	WORRELL, FRANK	
	Examin r	Art Unit	
	Shane F Gerstl	2183	

-- The MAILING DATE of this communication appears on the cover sheet with th correspondenc address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *24 December 2003*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *29 November 2000* is/are: a)☐ accepted or b)☒ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-20 have been examined.

Papers Received

2. Receipt is acknowledged of amendment paper submitted, where the paper has been placed of record in the file.

Response to Amendment

3. The drawing objections, claim objections, and 35 U.S.C. 112 (2) claim objection set forth in the action mailed 02 October 2003 have all been overcome by the filed amendment and are thus withdrawn.

4. The amendment filed 24 December 2003 is objected to under 35 U.S.C. 132 because it introduces new matter into the disclosure. 35 U.S.C. 132 states that no amendment shall introduce new matter into the disclosure of the invention. The added material which is not supported by the original disclosure is as follows: Claims 1 and 14 include a limitation where an instruction at a branch target address is fetched while an instruction sequential to the branch instruction is decoded. This limitation is not taught anywhere in the specification.

Applicant is required to cancel the new matter in the reply to this Office Action.

Response to Arguments

5. Applicant's arguments with respect to the suggested title have been fully considered and are persuasive. The suggested title has been withdrawn, however, the argument that the title is descriptive is not persuasive and the objection remains.

6. Applicant's arguments, see page 11, filed 24 December 2003, with respect to the section headings have been fully considered and are persuasive. The objection of the specification has been withdrawn.

7. Applicant's arguments with respect to claims 1-14 have been considered but are moot in view of the new ground(s) of rejection.

Specification

8. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

The following title is suggested: System and Method for Predicting Branches as Taken and Waiting for Evaluation of the Condition Before Prefetching the Next Instruction.

Drawings

9. The drawings are objected to under 37 CFR 1.83(a). The drawings must show every feature of the invention specified in the claims. Therefore, the decoding of a fall-through instruction while fetching the branch target address as described in claims 1 and 14 must be shown or the feature(s) canceled from the claim(s). No new matter should be entered.

A proposed drawing correction or corrected drawings are required in reply to the Office action to avoid abandonment of the application. The objection to the drawings will not be held in abeyance.

Claim Objections

10. Claims 17-20 are objected to because of the following informalities: the claims are dependent claims that depend on another dependent claim and are separated by claims that do not also depend from the same dependent claim. It should be kept in mind that a dependent claim may refer to any preceding independent claim. In general, applicant's sequence will not be changed. See MPEP § 608.01(n).

11. Claim 2 is objected to because of the following informalities: the preamble states that the method comprises plural steps, but it has been amended to comprise only one step. The 's' on the word "steps" simply needs to be removed.

12. Claim 4 is objected to because of the following informalities: The claim improperly refers to a claim not yet disclosed and that is not in the same claim grouping. 37 CFR 1.75 (f) and (g) shows that claims should be numbered consecutively and that all dependent claims should be grouped together.

Appropriate correction is required.

Claim Rejections - 35 USC § 112

13. The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

14. Claims 1-6 and 14-16 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

15. Claim 1 states the limitation, "decoding a second instruction stored at a next address adjacent said program counter address substantially simultaneously with said fetching." In the context of the claims, this limitation means that while an instruction stored at a branch target address is being fetched, an instruction sequential from the original branch instruction is decoded. The cited sections of the specification that supports the amendments nor the rest of the specification make any mention whatsoever of this decoding occurring substantially simultaneously with fetching the branch target instruction. Therefore, with no mention of the claimed invention in the specification one of ordinary skill in the art would not be enabled to make and use it.

16. In a similar manner, claim 14 states that an instruction at a branch target address is fetched at the same time as an instruction sequential to the branch instruction is decoded and thus the same argument given above applies to this claim.

17. Claims 1-6 and 14 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. As shown in the paragraph directly above, the disclosure does not teach the claimed invention presented by the limitations of claims 1 and 14. Therefore, the claimed limitations mentioned above are new matter and there is no indication that Applicant was in possession of such an invention at the time of filing.

18. Claims 1-6, 9, and 14-16 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

19. Claim 1 recites the limitation "decoding a second instruction stored at a next address adjacent said program counter address substantially simultaneously with said fetching " in lines 9-11. There is insufficient antecedent basis for this limitation in the claim. As shown above, the disclosure does not describe this aspect of the invention and thus the limitation is unclear in purpose, structure, and functionality.

20. Claim 14 recites the limitation "fetching a second instruction stored at a branch target address substantially simultaneously with said decoding (of an instruction sequential to the branch instruction)" in lines 5-8. There is insufficient antecedent basis for this limitation in the claim. As shown above, the disclosure does not describe this aspect of the invention and thus the limitation is unclear in purpose, structure, and functionality.

21. Claim 9 recites the limitation "said misdirect recovery address" in lines 5-6. There is insufficient antecedent basis for this limitation in the claim. The examiner is taking the claim to mean "said mispredict recovery address."

Claim Rejections - 35 USC § 102

22. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

23. Claims 1-4, 6-10, and 14-16 are rejected under 35 U.S.C. 102(b) as being anticipated by Trauben (5,509,130).

24. In regard to claim 1, Trauben discloses a method of conditional branching in a pipelined processor (figure 9 gives the flow of the pipeline stages), the method comprising the steps of:

a. fetching a first instruction stored at a branch target address in response to encountering a branch instruction, at a program counter address; Figure 10a shows that when a branch instruction (BNE) is encountered or decoded (d0 and d1), the first Target is fetched (T1). Figure 1, element 12 shows that a program counter exists in the system. Column 5, lines 35-46 show that the program counter is used for issuing instructions and its function does not need to be described because it is well know in he art. This well known in the art function is that the program counter points to the addresses of instructions that are fetched for execution.

b. decoding a second instruction stored at a next address adjacent said program counter address substantially simultaneously with said fetching; Figure 10a shows that a delay instruction and the sequential instruction after it are decoded while fetching T1. Column 18, lines 51-60 further explain that these are sequential instructions with respect to the branch (because of their placement in the sequential queue) that decode while the target is fetched. Since these instructions are sequential, they are adjacent to said program counter address of the branch.

c. and evaluating between taking a branch defined by said branch instruction and not taking said branch substantially simultaneously with said fetching. Column 18, line 61 shows that at time 6, the branch is resolved, meaning that it was being evaluated during time 5, or while the fetching was occurring. Figures 10a and 10b further show with arrows that the execute stage e0 determines the condition of the branch so the next instructions to be decoded are known and that this evaluation occurs at the same time as the fetching of the target T1.

25. In regard to claim 2, Trauben discloses the method of claim 1, as described above, further comprising the step of: fetching a third instruction stored at a sequential instruction address adjacent said branch target address in response to determining to take said branch; Column 18, lines 40-41 show that figure 10a illustrates the case when a branch is taken. The figure and column 18, line 43 – column 19, line 8 show that two instructions at the target address are fetched at once (Target 1 and 2) and that in response to the determination of the branch at e0, a third sequential address adjacent to the branch target address is fetched.

26. In regard to claim 3, Trauben discloses the method of claim 2, further comprising the step of: generating said sequential instruction address based upon said program counter address and a predetermined offset. As shown above, the program counter gives the address of the instructions. It is inherent that a sequential instruction has a predetermined offset that is added to the program counter address so that the instruction can be addressed. Figure 10a for example shows that two instructions are

fetched at a time and thus the offset for this case would be equal to address space that two instructions take up in memory.

27. In regard to claim 4, Trauben discloses the method of claim 2, further comprising the step of: generating said misprediction recovery address based upon an exception program counter address and a predetermined offset. As shown in figure 10b and column 19, lines 9-40, a sequential address is generated, and the instruction there is fetched. This instruction is executed when the branch is incorrectly predicted and is then a misprediction recovery instruction at a misprediction recovery address. As shown above, all instructions are pointed to by a program counter. It is inherent that a sequential instruction has a predetermined offset that is added to the program counter address so that the instruction can be addressed. Figure 10b for example shows that two instructions are fetched at a time and thus the offset for this case would be equal to address space that two instructions take up in memory.

28. In regard to claim 6, Trauben discloses the method of claim 1, as described above, further comprising the steps of:

a. generating a sequential instruction address adjacent said branch target address based upon said program counter address and a first predetermined offset. Column 18, lines 40-41 show that figure 10a illustrates the case when a branch is taken. The figure and column 18, line 43 – column 19, line 8 show that two instructions at the target address are fetched at once (Target 1 and 2) and that in response to the determination of the branch at e0, a third sequential address adjacent to the branch target address is fetched, and inherently the

address for it is generated. As shown above, the program counter gives the address of the instructions. It is inherent that a sequential instruction has a predetermined offset that is added to the program counter address so that the instruction can be addressed. Figure 10a for example shows that two instructions are fetched at a time and thus the offset for this case would be equal to address space that two instructions take up in memory.

b. generating a mispredict recovery address adjacent said next address based upon an exception program counter address and a second predetermined offset. As shown in figure 10b and column 19, lines 9-40, a sequential address is generated, and the instruction there is fetched. This instruction is executed when the branch is incorrectly predicted and is then a misprediction recovery instruction at a misprediction recovery address. As shown above, all instructions are pointed to by a program counter. It is inherent that a sequential instruction has a predetermined offset that is added to the program counter address so that the instruction can be addressed. Figure 10b for example shows that two instructions are fetched at a time and thus the offset for this case would be equal to address space that two instructions take up in memory.

c. generating said branch target address based upon a program counter address and an address displacement of said branch instruction; Trauben shows in column lines 16-19 that a branch target address is calculated or generated. Trauben also discloses the existence of a program counter as shown above. Column 18, lines 16-19 of Trauben show that the branch target address is

computed. Since the branch target addresses are computed and an absolute address is not taken, it is inherent that this address is generated from the current address (stored in the program counter) and an offset (displacement) that when added to the current address gives the destination.

d. fetching a third instruction stored at said sequential instruction address in response to determining to take said branch as shown above;

e. and fetching a fourth instruction stored at said mispredict recovery address in response to determining to not take said branch. Figure 10b shows that when the branch is determined to not be taken in stage e0, the sequential or mispredict recovery address is decoded and a sequential instruction after this recovery instruction is fetched.

29. In regard to claim 7, Trauben discloses a pipelined processor (figure 1) comprising:

a. A multiplexer (figure 3, element 62);

b. A circuit configured to present

i. A branch target address based on a branch instruction stored at a program counter address in prediction of taking a branch; Figure 1, element 12 shows that a program counter exists in the system. Column 5, lines 35-46 show that the program counter is used for issuing instructions and its function does not need to be described because it is well know in he art. This well known in the art function is that the program counter points to the addresses of instructions that are fetched for execution.

Since the multiplexer selects the address to fetch from, the target (T1) fetched as in figures 10a and 10b and column 18, line 43 – column 19, line 40, is presented to the selection means. Figure 3 shows the target address comes from the target queue 60.

ii. A sequential instruction address having a first value adjacent said program counter address; Figure 10a and the same section of text shows that when the taken prediction is correct, a next sequential target instruction is fetched (and thus sent to the multiplexer), which is adjacent to the address in the program counter when pointing to the original target address. This is the sequential instruction of the claim. The sequential addresses come from sequential queue 58, however, if fetched before the program counter is updated from the branch, the address will be in the target queue 60 as shown in the text.

iii. And a mispredict recovery address to said multiplexer substantially simultaneously. Figure 10b and the same section of text again show that if the branch is mispredicted, a sequential address from the instruction immediately after the branch instruction is fetched (and thus sent to the multiplexer). Since the instruction immediately after the branch is shown to be decoded and executed on an incorrect prediction, this instruction is a mispredict recovery instruction at a mispredict recovery address as are the instructions immediately after it. The sequential addresses come from sequential queue 58.

The figures show that these fetches occur in successive cycles and thus follow a reasonable interpretation of being presented to the multiplexer substantially simultaneous.

30. In regard to claim 8, Trauben discloses the pipelined processor of claim 7, as described above, wherein said circuit is further configured to present said sequential instruction address having a second value adjacent said branch target address to said multiplexer in response to determining to take said branch. As shown above, the sequential instruction address is adjacent to the branch target address.

31. In regard to claim 9, Trauben discloses the pipelined processor of claim 7 wherein said circuit comprises: a prefetch program counter for storing an address presented by said multiplexer among said branch target address, said sequential instruction address and said misdirect recovery address. Figure 3, shows that the sequential queue stores the output of a multiplexer 64, which outputs the same value as multiplexer 62 due to the same inputs and control signal. Therefore, multiplexer 64 is equivalent to multiplexer 62. This sequential queue is a prefetch queue that receives the instruction presented by said multiplexer chosen from a sequential instruction address (which is also a mispredict address as shown above) and a branch target address. It is inherent that the program counter of the figure receives the address for this instruction using an equivalent multiplexer as well to choose from the target, sequential , and mispredict addresses for these instructions and thus the program counter is the prefetch program counter.

32. In regard to claim 10, Trauben discloses the pipelined processor of claim 7 wherein said circuit comprises an instruction register for storing said branch instruction. The instruction cache 34 of figure 3 stores the instructions and thus also the branch instruction. The enclosed IEEE definition of "register" states that a register is a device capable of retaining information of the aggregate information in a digital computer and thus the instruction cache is an instruction register.

33. In regard to claim 14, Trauben discloses a pipelined processor (figure 1) comprising:

a. a means for decoding a first instruction stored at a next address adjacent a program counter address; Figure 10a shows that a delay instruction and the sequential instruction after it are decoded while fetching T1. Column 18, lines 51-60 further explain that these are sequential instructions with respect to the branch (because of their placement in the sequential queue) that decode while the target is fetched. Since these instructions are sequential, they are adjacent to said program counter address of the branch.

b. a means for fetching a second instruction stored at a branch target address substantially simultaneously with said decoding in response to encountering a branch instruction at said program counter address; Figure 10a shows that a branch target address (T1) is fetched simultaneously with the decoding of an encountered branch instruction.

c. and a means for evaluating between (i) taking a branch defined by said branch instruction and (ii) not taking said branch substantially simultaneously

with said fetching. Column 18, line 61 shows that at time 6, the branch is resolved, meaning that it was being evaluated during time 5, or while the fetching was occurring. Figures 10a and 10b further show with arrows that the execute stage e0 determines the condition of the branch so the next instructions to be decoded are known and that this evaluation occurs at the same time as the fetching of the target T1.

34. In regard to claim 15, Trauben discloses the method of claim 1, further comprising the step of: fetching a third instruction stored at a mispredict recovery address adjacent said next address in response to determining not to take said branch. Figure 10b shows that a third and fourth sequential (Sequential3 and 4) are fetched (fetch S4) in response to determining to not take said branch (as shown in column 18, lines 42-43). Since the mispredict recovery address is the address of the instruction sequential to the branch instruction, this third sequential instruction is adjacent the next address and is a mispredict recovery address.

35. In regard to claim 16, Trauben discloses the method of claim 1, further comprising the step of: storing said program counter address in a stage of said pipelined processor for at least two cycles. Figure 9 shows that the fetch stage comprises two cycles, f0 and f1. This means that the program counter address for addressing the two instructions that are simultaneously fetched is stored for two cycles.

Claim Rejections - 35 USC § 103

36. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

37. Claims 5, 12-13, and 17-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Trauben in view of Hennessy.

38. In regard to claim 5, Trauben discloses the method of claim 1, as described above, further comprising the step of:

a. generating said branch target address; Trauben shows in column lines 16-19 that a branch target address is calculated or generated. Trauben also discloses the existence of a program counter as shown above.

b. Trauben does not disclose that this branch target address is based upon a program counter address and an address displacement of said branch instruction.

c. Hennessy, on page 148 discloses the use of an offset or displacement to be added to some position stored in a register, which yields a branch target address. The program counter given by Trauben is a register because it holds a value in close proximity as do general-purpose registers.

d. Hennessy discloses that using his method to calculate the branch target address provides a large range of possible targets for the branch. Hennessy also teaches on last paragraph that by using the PC as the register, hardware convenience is established. This large range of addressing possibilities and hardware convenience would have motivate done of ordinary skill in the art to

implement the method of Hennessy in Trauben for greater address range and hardware convenience.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify Trauben's invention to generate the branch target address based upon a program counter address and an address displacement of the branch instruction as taught by Hennessy in order to have a wide range of addresses to branch to and for hardware convenience.

39. In regard to claim 12,

a. Trauben discloses the pipelined process of claim 7 wherein said circuit comprises: an exception program counter for storing an exception program counter address used upon determining not to take said branch. As shown above, when the branch is determined as not taken, a mispredict recovery address is fetched from. It has been shown to be the sequential instruction immediately after the branch instruction. Thus the program counter, which points to this instruction, is also an exception program counter that stores an exception program counter address.

b. Trauben does not disclose that the exception program counter is disposed in an execution stage of said pipelined processor.

c. Hennessy has taught on page 404, top paragraph, that a sequential PC is always computed for every instruction. The mispredict recovery address is simply a sequential address from the branch instruction address. On the table directly below, Hennessy has taught that the sequential instruction address is

generated using two source operands, one being the PC (program counter). Hennessy shows on page 506 the use of an exception program counter (EPC) disposed in the execute stage of the pipelined processor.

d. By always computing the sequential address from the branch instruction address in the PC using the method described by Hennessy, a uniform and simple method of sending control to the next instruction is established. Hennessy teaches on page 509 that the exception (a mispredicted branch in this case) is detected in the decode stage and handled in the execute stage. By handling it in the next clock cycle, one can be assured that the correct information will be present. This uniformity and simplicity coupled with integrity of data would have motivated one of ordinary skill in the art to modify Trauben's design to incorporate the method of generating misprediction recovery address given by Hennessy.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Trauben to include the sequential addressing method of Hennessy in order to have a uniform and simplistic design with a high integrity of data while generating a misprediction recovery address.

40. In regard to claim 13,

a. Trauben discloses the pipelined processor of claim 7, wherein said circuit comprises:

i. a prefetch program counter for storing said program counter address; Figure 3, shows that the sequential queue stores or prefetches

the sequential instructions received from the instruction cache. The program counter in the figure is shown to address these prefetched instructions and is thus the prefetch program counter.

ii. an instruction register for storing said branch instruction; The instruction cache 34 of figure 3 stores the instructions and thus also the branch instruction. The enclosed IEEE definition of "register" states that a register is a device capable of retaining information of the aggregate information in a digital computer and thus the instruction cache is an instruction register.

b. Trauben does not disclose an exception program counter for storing an exception program counter address used in generating said mispredict recovery address having a second value proximate said program counter.

c. Hennessy has taught on page 404, top paragraph, that a sequential PC is always computed for every instruction. The mispredict recovery address is simply a sequential address from the branch instruction address. On the table directly below, Hennessy has taught that the sequential instruction address is generated using two source operands, one being the PC (program counter). Hennessy shows on page 506 the use of an exception program counter (EPC) disposed in the execute stage of the pipelined processor that has a second value inherently proximate (in the vicinity) of the program counter.

d. By always computing the sequential address from the branch instruction address in the PC using the method described by Hennessy, a uniform and

simple method of sending control to the next instruction is established. Hennessy teaches on page 509 that the exception (a mispredicted branch in this case) is detected in the decode stage and handled in the execute stage. By handling it in the next clock cycle, one can be assured that the correct information will be present. This uniformity and simplicity coupled with integrity of data would have motivated one of ordinary skill in the art to modify Trauben's design to incorporate the method of generating misprediction recovery address given by Hennessy.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Trauben to include the sequential addressing method of Hennessy in order to have a uniform and simplistic design with a high integrity of data while generating a misprediction recovery address.

41. In regard to claim 17, Trauben in view of Hennessy discloses the pipelined processor of claim 13, wherein said circuit further comprises: an incrementor coupled to said prefetch program counter address and configured to generate said sequential instruction address from aid program counter address. As shown above and in figure 3 of Trauben, the prefetch program counter address is stored in the program counter. It is inherent that an incrementor exists to update the program counter to the next sequential instruction.

42. In regard to claim 18, Trauben in view of Hennessy discloses the pipelined processor of claim 13, wherein said circuit further comprises: an adder (i) coupled to both said prefetch program counter and said instruction register and (ii) configured to

generate said branch target address by adding said program counter address to an address displacement of said branch instruction. Column 18, lines 16-19 of Trauben show that the branch target address is computed. Since the branch target addresses are computed and an absolute address is not taken, it is inherent that this address is generated from the current address (stored in the program counter) and an offset (displacement) that when added to the current address gives the destination.

43. In regard to claim 19, Trauben in view of Hennessy discloses the pipelined processor claim 13, wherein said circuit further comprises: an incrementor coupled to said exception program counter and configured to generate said mispredict recovery address. It is inherent that an incrementor exists to update the program counters (including the exception program counter) to the next sequential instruction.

44. In regard to claim 20, Trauben in view of Hennessy discloses the pipelined processor of claim 13, wherein said exception program counter is coupled to said prefetch program counter to receive said program counter address. As shown in the caption on page 506 of Hennessy, the exception program counter saves the address of the instruction that caused the exception. Since the prefetch program counter contains the current instruction address, when the branch instruction generates the exception, the exception program counter is inherently updated from the prefetch program counter.

45. Claims 11 is rejected under 35 U.S.C. 103(a) as being unpatentable over Trauben in view of Eckner (6,044,460).

46. In regard to claim 11,

a. Trauben discloses the pipelined process of claim 7, wherein said circuit comprises: an exception program counter for storing an exception program counter address used upon determining to not take said branch; As shown above, when the branch is determined as not taken, a mispredict recovery address is fetched from. It has been shown to be the sequential instruction immediately after the branch instruction. Thus the program counter, which points to this instruction, is also an exception program counter that stores an exception program counter address.

b. Trauben does not disclose that the exception program counter is disposed in a decode stage of the pipelined processor.

c. Eckner shows in figure 3 the use of an exception program counter disposed in the decode stage (element 408) of the pipelined processor. Column 6, line 66 – column 7, line 1 states the explicit coupling of the EPC in the decode stage.

d. The exception program counter placement taught by Eckner allows for the use of the handling of an exception as soon as possible. This means that fewer pipeline cycles need to be flushed than if the exception program counter was placed later in the pipeline. This advantage of time saved would have motivated one of ordinary skill in the art to modify Trauben's design to incorporate the placement of the exception program counter as described by Eckner.

It would have been obvious to one of ordinary skill in the art at the time of invention to modify the design of Trauben to place the exception program counter in the decode

stage as taught by Eckner so that the fewer pipeline stages need to be flushed and time is saved.

Conclusion

47. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

48. The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

49. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. The cited references given in the previous Office action remain pertinent and are cited with this action as well.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Shane F Gerstl whose telephone number is (703)305-7305. The examiner can normally be reached on M-F 6:45-4:15 (First Friday Off).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (703)305-9712. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Shane F Gerstl
Examiner
Art Unit 2183

SFG
March 19, 2004

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100